



# University of Central Punjab

## Faculty of Information Technology

### Data Structures and Algorithms

#### Fall 2021

Lab 06	
Topic	<ul style="list-style-type: none"><li>• LinkedList</li></ul>
Objective	<ul style="list-style-type: none"><li>• The basic purpose of this lab is to implement ADT of Linked List and test its applications.</li></ul>

#### Instructions:

- Indent your code.
- Comment your code.
- Use meaningful variable names.
- Plan your code carefully on a piece of paper before you implement it.
- Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp
- **void main() is not allowed. Use int main()**
- **You have to work in multiple files. i.e separate .h and .cpp files**
- **You are not allowed to use system("pause")**
- **You are not allowed to use any built-in functions**
- **You are required to follow the naming conventions as follow:**
  - o **Variables:** firstName; (no underscores allowed)
  - o **Function:** getName(); (no underscores allowed)
  - o **ClassName:** BankAccount (no underscores allowed)

Students are required to complete the following tasks in lab timings.

## Task 1

Create a C++ generic abstract class named as LinkedList with the following:

### Attributes:

- ✓ Type Data;
- ✓ Node <Type> \*head;
- ✓ Node <Type> \*tail

### Functions:

**virtual void insertAtFront(Type) = 0;**

Adds the element of type Type at the head of the linkedlist.

**virtual void insertAtEnd(Type) = 0;**

Adds the element of type Type at the tail of the linkedlist.

**virtual Type removeFromFront() =0;**

Removes and returns the first element of the linked list, and reduces size of the linked list by 1.

**virtual Type removeFromEnd() =0;**

Removes and returns the element at the tail of the linked list, and reduces size of the linked list by 1.

## Task 2

Modify the code done in class and implement the linked list using **Head** and **Tail** pointers.

### Functions of Linked List:

**front()** – Returns the value of the first element in the linked list.

**back()** – Returns the value of the last element in the linked list.

**insetAtFront(int)** – Adds a new element at the beginning of the linked list.

**insetAtEnd(int)** – Adds a new element at the end of the linked list in O(1).

**removeFromFront()** – Removes the first element of the linked list, and reduces size of the linked list by 1.

**RemoveFromEnd()** – Removes the last element of the linked list, and reduces size of the linked list by 1.

**empty()** – Returns whether the list is empty(1) or not(0).

**size()** – Returns the number of elements in the list.

### Task 3

Using the linked list made in task 2, create a function insertSorted(), which should insert the data in the list in a sorted manner. Use tail pointer.

### Task 4

Create a function to reverse the linked list.

Example:

Initial Linked List : 1 -> 2 -> 3

Reversed Linked List: 3 -> 2 -> 1

After reversing, if you perform an insert operation, it should be inserted next to 1.